

# Pointers for JS Project

## LocalStorage

O **localStorage** vai servir como a vossa base de dados, como tal tem que enserir, obter e apagar dados do mesmo.

Visto que este tema não foi falado em aula, segue alguns “pointers” para o mesmo.

### Set (Inserir)

Para inserir dados no local storage temos que associar uma key (chave) ao valor (value).

```
localStorage.setItem(key, value)
```

```
var email = "joao.campanhas@gmail.com"  
localStorage.setItem("email", email)
```

### Get (obter, ir buscar...)

Para obtermos um valor previamente guardado no local storage, temos que utilizar a key pelo qual o guardamos

```
localStorage.getItem(key)
```

```
localStorage.getItem("email") //"joao.campanhas@gmail.com"
```

### Remove (Apagar)

A accao de remover é semelhante à de get, apenas necessitamos de usar a funcao `removeItem` com j.a key e o valor e completamente apagado do `localStorage`.

```
localStorage.removeItem(key)
```

```
localStorage.removeItem("email")
```

## Clear (Apagar todos)

Clear remove todos os dados inseridos no localStorage.

```
localStorage.clear()
```

## Outros pointers

### Objetos e Arrays

O local storage não salva por norma objetos ou arrays só strings, como tal para ser possível guardar um objeto ou array temos que o serializar primeiro e ao obter os valores deserializar.

A função `Json.stringify` permite transformar um objeto em string e a função `Json.parse` permite realizar o inverso, tornando um string que tenha a formatação de um objeto, em um.

```
let user = {
  username: "nuno",
  password: "password"
}

localStorage.setItem("user", Json.stringify(user))
var getUser = Json.parse(localStorage.getItem("user"))
```

## Possível Estrutura de Dados

Os excertos de código apresentados nesta secção são apenas pseudo código.

De acordo com os requisitos do projeto, eu proponho a seguinte estrutura de dados.

Todos os utilizadores registados devem ser guardados numa lista do objeto user, desta forma ao realizar o Login temos uma lista pelo qual podemos confirmar se o utilizador existe.

```
"users" : [ { user } ]
```

O localStorage deve também ter/saber que user está logado, como tal deve também existir um loggedUser, que representa o utilizador logado, este dado deve apenas conter o objeto do user, e deve ser atualizado sempre que for realizado o Login e devido às especificações do projeto também após o registo, visto que após o registo o utilizador é redirecionado para a Home page.

```
"loggedUser" : { username }
```

Equivalentemente aos users registados, deve também existir um localStorage de shifts onde serão inseridos de todos os shifts criados.

De modo a poderem ser pesquisados num dos pontos do projeto.

```
"shifts" : { shift }
```